specification") and further in view of Hollberg et al. This rejection is respectfully traversed.

Independent claim 1 specifically recites that the *client object is configured to identify methods of said target object which are remotely accessible and support manipulation of properties of said target object.* Independent claim 7 specifically recites that the *client object identifies methods of said target object which are accessible remotely.* Independent claim 18 specifically recites that the *client object is configured to identify remotely accessible methods of the target object that support programmatic manipulation of target object properties by processes in the second virtual machine.* The Action asserts that Hill teaches a client object that is configured to identify methods of the target object that are remotely accessible, and cites col. 7, line 64 through col. 8, line 20 in support of this assertion. Applicants disagree. The cited text reads as follows:

> FIG. 4B is a block diagram illustrating the marshalling of a cell object to the client. When a client wants to retrieve the formula of cell A1 represented as cell object 404, the client executes the following statements.

```
plBasic.fwdarw.GetCell("A1", pCell);
formula = pCell .fwdarw. GetFormula();
```

> The spreadsheet proxy 407 is pointed to by pointer plBasic 408. The client first invokes the spreadsheet proxy method GetCell. The method GetCell packages the method name "GetCell" and the string "A1" into a message and sends the message to spreadsheet stub 406. The spreadsheet stub 406 unpackages the method name and string. The spreadsheet stub 406 then invokes the GetCell method of the spreadsheet object 403. The method GetCell returns to the spreadsheet stub 406 a pointer to cell object 404. The spreadsheet stub 406 then marshals the cell pointer by creating cell stub 409 for cell object 404, assigning a message address to cell stub 409, packaging the message address and an unmarshal class identifier (described below) into a message, and sending the message to the spreadsheet proxy 407. When the spreadsheet proxy 407 receives the message, method GetCell then unmarshals the pointer to the cell object 404.

Nothing in the cited text discloses or suggests identifying methods of a target object which are remotely accessible. Therefore, Applicants submit that the rejection of claims 1, 7, and 18 is improper and should be withdrawn. If the Examiner refuses to withdraw the rejection, then Applicants request the Examiner specifically describe what portions of the cited text disclose or suggest identifying methods of a target object which are remotely accessible, as recited in the claims.

Independent claims 1 and 18 each recite *registering said target object and a network adaptor for a network protocol with a framework at said remote station*. Similarly, independent claims 15 and 17 each recite that *said at least one target object and said at least one network adapter being registerable with a framework at said first machine*. The Action asserts that the Corba specification (referred to in the Action as the OMG) teaches these limitations, citing page 32. Applicants disagree. The text of page 32 reads as follows:

> Fig. 6 on page 32 shows how interface and implementation information is made available to clients and object implementations. The interface is defined in IDL and/or in the Interface Repository; the definition is used to generate the client Stubs and the object implementation Skeletons.
>
> The Object Implementation information is provided at installation time and is stored in the Implementation Repository for use during request delivery.
>
> 3.1.1 Object Request Broker
>
> In the architecture, the ORB is not required to be implemented as a single component, but rather it is defined by its interfaces. Any ORB implementation that provides the appropriate interface is acceptable. The interface is organized into three categories:
>
> 1.   those operations that are the same for all ORB implementations
>
> 2.   those operations that are specific to particular types of objects
>
> 3.   those operations that are specific to particular styles of object implementations.

Nothing in the cited text discloses or suggests *registering said target object and a network adaptor for a network protocol with a framework at said remote station*, as recited in independent claims 1 and 18. Similarly, nothing in the cited text discloses or suggests *at least one target object and said at least one network adapter being registerable with a framework at said first machine*, as recited in independent claims 15 and 17. Therefore, Applicants submit that the rejection of claims 1, 15, 17, and 18 is improper and should be withdrawn. If the Examiner refuses to withdraw the rejection, then Applicants request the Examiner specifically describe what portions of the cited text disclose or suggest registering said target object and a network adaptor for a network protocol with a framework, as recited in the claims.

In addition, the Action asserts that page 8 of the CORBA specification teaches that "[t]he Object Implementation information is provided at installation time and is stored in the Implementation Repository for use during request delivery. Applicant notes that the Examiner has not provided Applicants with a copy of page 8 of the CORBA specification. Therefore, this rejection is improper.

C. Conclusion

In view of all of the above claims 1-18 are believed to be allowable and the case in condition for allowance which action is respectfully requested.

No fee is believed to be required by this response as determined on the accompanying transmittal letter. Should any other fee be required, please charge Deposit 50-1123. Should any extension of time be required please consider this a petition therefore and charge the required fee to Deposit Account 50-1123.

Respectfully submitted,

Date: February 19, 2003

BY: _____

Jed W. Caven #40,551
HOGAN & HARTSON LLP
One Tabor Center
1200 17th Street, Suite 1500
Denver, Colorado 80202
Phone: (303) 454-2432
Fax: (303) 899-7333

5